DOCUMENT RESUME

ED 262 753 IR 011 831

AUTHOR	Sherwood, Robert D.
TITLE	Models of Computer Use in School Settings. Technical Report Series, Report No. 84.2.2.
INSTITUTION	George Peabody Coll. for Teachers, Nashville, TN. Learning Technology Center.
PUB DATE	84
NOTE	38p.; To appear as a chapter in "Computers in Education from Literacy to Applications," by C. Kinzer, R. Sherwood, and J. Bransford. Holt, Rinehart & Winston, 1985.
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; Computer Managed Instruction; Computer Simulation; Drills (Practice); Instructional Innovation; Man Machine Systems; *Microcomputers; *Models; Problem Solving; Programed Tutoring; Programing; *Skill Development; Teaching Methods
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

Designed to focus on student learning and to illustrate techniques that might be used with computers to facilitate that process, this paper discusses five types of computer use in educational settings: (1) learning ABOUT computers; (2) learning WITH computers; (3) learning FROM computers; (4) learning ABOUT THINKING with computers; and (5) MANAGING learning with computers. Within each section, the method is first discussed in general, then specific examples of software that could be used with each method are described. At the end of each section, three questions are posed and answered that address reasons why the utilization would be useful to students, when the utilization should be considered, and what kinds of skills it would provide. Specific reference is made to such topics as computer literacy, computer literacy courses, development of programming skills, drill and practice software, tutorial programs, simulation software, computer games, writing and problem solving tools, administrative uses for computers, and communication between computers in a school or with machines at other locations. A list of references completes the document. (JB)

Models of Computer Use in School Settings

Robert D. Sherwood, Ph.D.

Associate Director

Learning Technology Center

Peabody College of Vanderbilt University

Nashville, TN 37203

To appear as a chapter in Kinzer, C., Sherwood, R. & Bransford, J. Computers in Education from Literacy to Applications. Holt, Rinehart & Winston, 1985.

Learning Technology Center Technical Report Series

Report No. 84.2.2

CHAPTER SEVEN

MODELS OF COMPUTER USE IN SCHOOL SETTINGS

A variety of terms have been used to describe how computers might be used in schools. The most general and oldest term is CAI for Computer Assisted Instruction. Other terms that have appeared in the literature are CMI (Computer Managed Instruction) and CBI (Computer Based Instruction). A common term in European studies is Computer Based Learning (CBL). While some differences exist in these terms, (e.g. CMI is generally thought of as using computers to assist teachers in managing the instructional offerings that students work with as opposed to the more general term CBI), they all share a common thread of emphasis on the computer focusing the instruction.

The approach of this chapter will be somewhat different from typical descriptions of the ways that computers can be utilized in schools. The chapter will focus on student learning and illustrate some of the techniques that might be used with computers to facilitate this learning. This view is not unique to this text. Other authors such as Taylor (1) and Luehrman (2) have adopted this view in their work. Taylor has used the terms "Tutor", "Tool", and "Tutee" when describing the way computers might be used in the schools. As these terms imply, Taylor believes that the computer can be used as a "Tutor" to teach students, as a "Tool" for students to use as they would other educational tools, and as a "Tutee" that students can teach.

Luehrman has expressed these same ideas is a slightly different way. He also sees three roles. They are "learning about the computer", learning with the computer", and "learning from the computer". The approach of this chapter will extend the three utilizations that Luehrman has proposed to five. These are:

1. Learning ABOUT Computers

2. Learning WITH Computers

3. Learning FROM Computers

4. Learning ABOUT THINKING with Computers

5. MANAGING learning with Computers

This chapter discusses each of these five methods of use in separate sections. Within each section the method is first discussed in a general manner , then specific examples of software that could be used with each method are described. At the end of each section, three questions are posed and answered. These are:

1. WHY would this utilization be useful to students and teachers?

2. WHEN would this utilization best be considered?

3. WHAT kinds of skills does it provide?

I. LEARNING ABOUT COMPUTERS

In a recent discussion of learning about computers, several of the authors of this text reflected on their first experiences with computers. For most of us the first computer we ever saw was a large mainframe machine that took up at least one whole floor of a building with special air conditioning and security requirements. We never, of course, actually touched the computer. It was too delicate for students to even get close to. Information to be processed by the machine was punched on computer cards and handed to a person who fed them into a card reader that was connected to the computer. After several hours or sometimes a full day a printout of the program submitted would be placed in a box for pickup. Many times a small mistake such as misplaced period would cause the entire program to fail and the time consuming process would have to be repeated. Needless to say, this first experience was not especially reinforcing and many expressed dislike of computers.

The situation today is very much different. As you have seen in previous chapters, there has been a major change in the computer hardware that most students utilize for the first time. While large mainframe computers are still used, and will probably still be needed for several years to come, the growth of microcomputers allows students to have a completely different type of experience than students as few as ten years ago might have had. Students now can work with a computer on a much more personal, one to one basis. The machine itself is not locked in some forbidden room never to really be seen. Individual students or groups of students can work directly with a machine that may have almost as much computer "power" as the large machines of several years ago. While programming errors, including missplaced periods, will still cause programs to stop, the feedback time is almost instantaneous in many current systems. In addition, changes in information storage capabilities (floppy disk drives vs hard disk drives that were never seen), output devices (screens vs only printouts), and the availability of small fast printers offer a chance for a much richer first experience with computers.

There are a variety of ways that schools could implement this new richer environment through "learning about computers". Our recommendations center around answering the question, "What level of knowledge about the computer do students, at the grade level of interest, need to have to appropriately use the machine?."

The most basic level of knowledge is what is often termed "Computer Awareness". This is probably best described as teaching students enough about a computer system so that they could use the system as a learning center. For example, not a great deal of computer knowledge is needed to physically operate a current microcomputer. The sequence necessary to turn the machine on and load a program that has already been written is rather simple. Individual teachers from the primary grades on up can be taught how to do this type of task and can easily teach it to their students. If this is to be a major use of the machine in a classroom then the students in that room may not need to be involved in much "Learning About Computers". Ideally, however, some knowledge above this beginning level would be useful. Some development of the idea that computers are not "magic" or really "intelligent" would be useful for a teacher to communicate to the students. Nevertheless, students at early grade levels are probably not interested in, or really capable of, understanding such concepts as "bits" or the workings of microprocessors.

A second level of learning about computers involves what is commonly termed a "Computer Literacy" program or course. Included in this type of program are some of the topics covered in this text in Chapters 2 and 3. The class will usually cover topics such as:

1. A beginning awareness of what a computer system is and the components that make up a system.

2. The languages that are used with computers.

3. The basic operation of a computer.

4. The impact of computers on our society and on particular occupations both now and in the near future.

5. An introduction to the actual programming of a computer.

6. Utilizing the computer as a tool such as with word processing.

As can be seen from such a listing, "Computer Literacy" involves what was covered in "Computer Awareness" and then extends this information into other areas. This type of course or program has been most commonly implemented in the middle or junior high schools. Students of this grade level can probably be expected to understand more of the terminology that accompanies computer utilization as well as some of the higher level concepts. For example, the concept of information flow from a disk drive to a computer, the idea that computers operate in another arithmetic base (base 2 or binary), and some of the concepts of computer programming are all areas where reasonable success with this age student would seem reasonable.

As can be seen from the list of topics above, two types of computer software are used in computer literacy courses. First, a computer language must be chosen in order for topic 5 (an introduction to the actual programming of a computer) to be implemented. As we saw in chapter 4, some controversy exists about which programming language should be used with students in their first experience with computers. Again, the age of the student should be considered when this situation arises. The language BASIC contains statements in which variables are used in expressions that may not make logical sense to students. One such commonly used statement is LET X=X+1. This expression means that the new value of variable X is equal to its old value plus one. This is not an easy concept to follow for many students. In fact, authors such as Luehrman(3) have suggested that the concept of variables be placed at the end of a course in which BASIC is introduced. Luehrman's book Computer Literacy: A Hands On Approach was designed for the junior high school audience and presents one of the better approaches to teaching BASIC for that age student. However, learning BASIC is generally not recommended for elementary school children. The Logo language ,also mentioned in chapter 3, and further discussed in the "Learning About Thinking with Computers" section of this chapter, is an alternative.

Other software that may be used in a computer literacy course involves the utilization of "tool" type software such as word processing packeages. Such applications software would be used to introduce the computer as a method to assist the student in their other classes and as a foundation for later computer work.

A third level of learning about computers is the development of computer programming skills. Students at a high school level should have the opportunity to take courses that extend programming skills that they have learned in previous courses. The current use of Pascal as the language of the SAT advanced placement examination for computer science necessitates that students have the possibility to work with such a system in a course. Many microcomputer systems have Pascal systems available as options. As noted previously, some disagreement about the use of BASIC in programming classes has been noted. Some university computer science faculty have strong feelings that experience with BASIC, no matter how it is taught, is not helpful for their students and prefer that students do not take it. Other authors contend that, properly taught, BASIC is not an impediment to learning other languages. A discussion of both languages is presented in chapter 3.

Questions to Consider

Why would this utilization, i.e., "learning about computers" be useful to students and teachers?

Much has been written in the popular press about the need for "computer professionals" in the coming years. While it is true that large *percentage* increases for computer programmers and systems analysis will occur, the actual number of these jobs is not especially large compared to the total labor market. Many people will need computer <u>skills</u> in order to manipulate the information that is growing rapidly in the country. They many not, however, need to know how to program a computer. With this in mind, it appears that all students need to be at least on the first level of this utilization that of "awareness". While many need to progress to the second level i.e., "literacy", relatively few will need the skills of the third i.e., "programming".

<u>When would this utilization best be considered?</u>

As was mentioned previously, the "when" question appears to be tied to the developmental level of the child. "Computer Awareness" can be started at a very early age, with students adding skills to move to the literacy level at later grade levels. We also noted that, while some controversy over appropriate grade levels for introduction of various languages exists, it appears that languages such as BASIC and Pascal need to be deferred until at least the junior high school years. Advanced programming classes seem to be most appropriate at the high school level.

<u>What kinds of skills does the utilization provide?</u>

The skills will, again, depend on the level of implementation. Each level builds on the previous with "Awareness" providing skills in using the machine, "Literacy" adding basic understanding of the machine and some programming concepts, and finally "advanced programming" which involves looking at higher level languages and the concepts that they contain.

II. LEARNING FROM COMPUTERS

This utilization of computers is perhaps the most familiar to the general public. Here, computers are used to either "tutor" a student (tutorial software) or to provide additional practice on specific skills (drill and practice software). A general way to visualize the instructional process that takes place is represented by the diagram in Figure 7.1.

Figure 7.1 About Here

As can be seen in the figure, the instructional process provides information, asks questions, judges responses, provides feedback, and either provides hints/remediation or moves on to the next instructional sequence. In drill and practice situations, the first block of "presents information" is usually not implemented. The student has usually received instruction from another source and practicing the skill through questions provided by the computer. A tutorial program will have each of the blocks in the figure although the depth to which each task is carried out varies a great deal.

We utilize the term "Learning From Computers" to describe this method because, in many instances, the flow of information is largely undirectional. It flows <u>from</u> the computer <u>to</u> the student with the student making relatively limited responses to the proposed questions. While we are concerned that this almost unidirectional information flow may restrict the utility of this use of computers, it may be appropriate for some situations. In order to explore this possibility we shall provide some examples of computer software that illustrates this approach.

Drill and Practice

Drill and practice software dominated early computer learning situations and, today, is still offered by many companies in a variety of forms. For example, consider the program "Speed Drill" prepared by the Minnesota Educational Computer Consortium (MECC)(4). Beginning screen displays are shown in figures 7.2 and 7.3.

Figures 7.2

After asking for the student's name (we will use the name "Jim" in our example) and asking whether instructions are desired, the screen shown in figure 7.2 appears. Students are given the opportunity to work with any of the basic arithmetic areas, to choose the difficulty level of the problems, the speed of presentation and the number of problems. In this way the program can be used by a variety of students at different ability levels. After the students have made their choices, problems such as the one illustrated in Figure 7.3 are presented. The problem in the Figure 7.3 was from the "moderate difficulty" problem set.

Figure 7.3 about Here

A correct student response results in a simple reinforcement phrase such as "good job, Jim" being displayed at the bottom of the screen, while an incorrect response yields "No, that is not correct. Try again". The student is given a second chance to solve the problem. If the student does not solve the problem correctly the answer is given after the second trial. The student has only a limited amount of time to respond to each of the problems (between 6 and 10 seconds) after which the computer indicates that too much time was taken and that the student should try again.

After completing the number of problems selected, the student reviews a summary screen that provides information on the number of problems attempted, number correct on the first presentation, number correct on the second presentation, and the number missed. In addition, a challenge score is displayed that is based upon the speed level chosen, difficulty level of the problems, and number correct.

It is useful to consider more closely some of the student/computer interactions that take place when a program such as the one just discussed is used. First, some decisions must be made by the student about the appropriate type of problem and difficulty/speed levels. Ideally, the classroom teacher provides some guidance for these decisions based upon current classwork or the student's specific needs.

Next, the student is given a problem and asked to calculate the correct answer. In this phase the computer is really not doing much more than acting as a paper worksheet. No real "computer power" is used, although in this specific example some serious programmer work was involved in producing large numbers on the screen. Even so, a problem with the screen display of "Speed Drill" is noticeable when it is fully considered. The practice problem shown in Figure 7.3 is presented horizontally not vertically as is normally done on paper. This is a disadvantage of this program and is also noted in many similar programs. Many students might feel compelled to re-write the problem with pencil and paper in order to solve it. Other mathematics programs, some of which are discussed in Chapter 6, use alternative displays that are probably more appropriate.

Returning to our consideration of the interactions between computer and student, it is noteworthy that most drill and practice programs follow a feedback mechanism similar to the one used the above example. After two (or sometimes three) attempts the student is provided with the correct answer. Some alternatives to this procedure are (1) to not provide the correct answer until the total session is completed and then to display the problems missed or (2) to provide hints for students and then present additional problems to be solved that are similar to the ones that the student missed.

Attemps to use student responses in a more "intelligent" manner than in the previous drill and practice example, has been the subject of much discussion. Recent work by Brown and Burton (5) on such programs as "Buggy", where the program diagnoses students' difficulty in math problems based upon a very large set of rules in the program is most interesting. Work in this area has sometimes been called "Intelligent Computer Assisted Instruction" (ICAI) because of the higher level of interaction that goes on between the student and the computer system. Many researchers in the artifical intelligence area are actively developing and testing a variety of systems that utilize "expert systems" for diagnosis of student responses in many content areas. For a review of several projects and the field as a whole see Suppes (6).

The use of "challenge scores", "goals", or mechanisms that allow the student to "play against the computer" or against other students are all methods by which motivation can be increased. Some students find such competion rewarding, especially if they "beat" the computer or score the highest in their class. Other students, however, especially if they are <u>not</u> successful, may be "turned off" by the use of such programming. A teacher may need to consider carefully if the use of "scores" will be helpful to teacher's students. Malone (7), in his work with computer games, has found that having scores kept by the computer did increase the popularity of games over identical games with no score. Our concern is that scores not be used as a method that would <u>decrease</u> the self-confidence of students who do not do especially well on a particular program.

Tutorial Programs

Tutorial programs are in many ways similar to the drill and practice programs outlined in the previous section. The major differences center around the use of the computer to deliver the "primary" instruction on a topic through the use of information screens. This differs from drill and practice programs in which initial learning took place elsewhere, with the computer only being used as a practice medium. As we have noted, there is some concern about whether the computer is really the best vehicle to deliver this type of textual information. Early work in this area often presented screen after screen of what was essentially text with little graphic additions or choices for students in terms of alternative pathways. More recent work has been much more mindful of the importance of graphics and student control as important variables in the presentations of new concepts.

The line between tutorial software and simulation software (which will be discussed in the next major section), is becoming less distinct. Some developers of programs that cross both areas, such as Dr. Albert Bork of the University of California at Irvine, describe their programs as "Dialogues" (8) in which information is presented to students in a variety of ways: simulation, graphic, and textualmofrd. Student responses are also used to enhance the program by providing differing presentations to students of various abilities. A student who has difficulty on a particular part of the program might see a section of the program, sometimes called a branch, that presents the same material in an different manner.

The program discussed below also makes use of various features of the computer to present material. The screen shown in Figure 7.4 is from a program titled "Plant Growth" prepared by Classroom Consortia Media, Inc. (9). This program uses a variety of methods to present information.

Figure 7.4 About Here

The mixed text and graphics of Figure 7.4 highlight key terms of the topic presented. Simulations of actual experiments showing the growth of plants under various conditions utilize previously presented material in an alternative manner. After both informational and simulation screens have been completed the student can check their knowledge of the concept by a quiz.

While it is easy to say that tutorial programs should be highly graphical, branched, and interactive, it is not an easy task to produce such software. The process of creating branches, alternative instructional sets for students of various ability levels, is extremely time consuming. Student misconceptions and difficulties must be predicted and appropriate instruction developed to assist the student. Development time estimates vary, but numbers on the order of 200-300 development hours per program for just pilot programs are not at all unreasonable. These first attempts must then be field tested and refined in order to produce a final product.

Some developers, such as Bork's group, use the language Pascal for their program development. Others use authoring languages such as Pilot, Private Tutor (IBM), McGraw-Hill Interactive Authoring System, and similar products for their work. The interesting feature of some of these authoring languages (and also Pascal) is the ability to control external devices such as video tape players, video disk players, and random access slide projectors from within the language. This dimension, which is still somewhat in the experimental stage, offers some interesting possiblities for tutorial and other types of software. For example, the ability to insert single scenes or more extended visuals under computer and learner control could result in the student being allowed to experience situations of much greater depth than is currently possible with computer graphics. If students were able to see a video segment of an experiment that they had just simulated with the computer, rather than simply a table of expected values, the learning experience might be significantly improved. Increased costs, and the extra time to produce good video segments, are both factors that will slow the growth in this area. Some of the possible applications of this technology are covered in more detail in Levin (10).

Questions to Consider

Why would this utilization, i.e., "Learning From Computers" be useful to students and teachers?

Drill and practice software gives the student an opportunity to obtain additional practice on skills that are needed for a particular concept. Many times these skills are of a fairly "low" level; for example, arithmetic facts that must be learned for the student to proceed to more complex work. It is unlikely that a drill and practice format will be useful in developing students' thought processes in such domains as analysis, synthesis, or evaluation. Tutorial programs may, if properly developed, allow students to explore areas of thinking above the knowledge level. The production of tutorials that use a variety of information presentation and feedback mechanisms are more likely to create such possiblities.

When would this utilization best be considered?

Clearly, even very young children can interact with drill and practice programs to work on specific skills. The more important consideration should be on the congruence of the program with what is being done in the class. If the software is using a teaching strategy different from that of the teacher, confusion among the students might occur. Ability levels of the students must also be considered; students presented with tasks either too hard or too easy can easily loose interest in using the computer. Ability of the student to read the text on the screen, even if it is kept to a minimum, must also be considered. Tutorials may act as remediation, initial instruction, or enrichment of classroom work, but a clear knowledge of the role they are to take must be known to the student.

What kinds of skills does the utilization provide?

As previously mentioned, drill and practice will be somewhat limited to the level of skill and concept development that these types of programs can provide. Expectations of high level objectives are probably unrealistic. Tutorial programs vary widely, with newer programs being able to present material at much higher levels, especially when simulation or gaming strategies are used along with information presentation.

III. LEARNING WITH COMPUTERS

In the previous section we expressed concern that many instances of drill and practice and tutorial software have a unidimensional information flow. The information goes from computer to student. The student makes rather limited responses back to the computer. The utilization that will be discussed in this section differs in that we see a more equal partnership between computer and student. The student will either be (a) making decisions about how he or she wants the computer to interact with a simulation or game, or (b) using the computer as a tool to shape information that is already possessed. As before, the dividing line is not sharp between one use of the computer and others, but generally this utilization can be differentiated from the previous two sections based upon the types of learning that is taking place.

Simulations

Simulations have been utilized for several years outside of education, with and without computers, in military, business, and governmental settings. They have enjoyed popularity because they place the user in a setting where decisions similar to those in a person's environment must be made. This gives the participant the opportunity to practice skills in situations where wrong decisions do not have disastrous results. For example, simulation training for airline pilots has been used for many years to provide a variety of situations without endangering people or equipment.

In education, simulations have become increasingly popular, especially in science, mathematics, and the social sciences. In science, many situations in the biological sciences cannot be done in a lab or in short time periods in the field. Simulations give studenta the chance to experience situations not normally available in classroom settings.

A good example of a science simulation is "Odell Lake", developed as an ecological simulation for grades 4 through 6 by the Minnesota Educational Computer Consortium (MECC) (11). In this simulation, the student plays the role of a particular type of fish that lives in Odell Lake. The student is presented with a variety of situations that might occur in nature such as other fish coming into the student's territory; various types of food being make available; and the entrance of non-fish predators such as an otter. Figure 7.5 shows a screen display of "Odell Lake". The "student-fish" (left side of the screen in Figure 7.5) must make decisions about how to react when these changes in the situation occur. The decisions are considered by the computer program which then lets the student know if whether or not the response was appropriate and then introduces a new change in the situation.

Figure 7.5 About Here

In Figure 7.6 the student has made an incorrect decision about whether to ignore a certain kind of fish and the predator fish (right side of screen) is coming in with open mouth and giant teeth to eat the student-fish.

Figure 7.6 About Here

Two important positive characteristics of a good simulation, both of which are exhibited in Odell Lake are: (1) the simulation allows students to make mistakes for which the consequences are non-threatening, and (2) it allows students to exercise some control over the learning process. Many times our educational situations are "right answer only" where students are strongly encouraged not to make mistakes. However, if schooling is really intended to mirror the real world, we should allow students more opportunities to make mistakes and to learn from them. Learning why something is wrong, and then being able to repeat the process with this experience as a guide, often results in stronger learning than just having students trying to get the correct answer to a question from facts that have been memorized. Simulations allow students to try different approaches in response to a given situation without having to fear the consequences of an incorrect action. In terms of control, simulations allow students to become decision makers rather than simply being receivers of information. It appears that this type of control is intrinsically motivating to students and helps to make the simulation an effective learning tool.

Computer Games

A computer game differs from a computer simulation in that the game does not necessarily have to model a real world situation. In addition, learning that takes place in the game is usually picked up indirectly as the student attempts to develop a game winning strategy. A good example of a computer learning game is "Hurkle", also developed by MECC. The objective of the game is for the student to find the Hurkle (a small cartoon type character) after the Hurkle has hidden somewhere on the screen. Figure 7.7 shows a screen display from "Hurkle" in which Hurkle is hiding on a 10 by 10 grid. The program randomly hides the Hurkle in an appropriate position on the screen. The student uses the numbers representing the different positions to guess where the Hurkle is located. If the guess in incorrect, the computer gives the student a hint by printing something like "Go Southwest". The student then uses this information to try a second guess. This continues, with each incorrect answer eliciting a new hint, until one of the guesses is correct. In this game, the student is learning about representing positions on a line or a two dimensional grid in terms of numbers. This is dome not as an explicit part of the game but as a necessary part to find the Hurkle.

Figure 7.7 About Here

Simulations and games, while perhaps a better and more effective use of the computer's capabilities as an educational tool than many drill and practice programs, do have limitations. In order for a simulation to be effective and useful it must present an accurate model of the process or situation it is supposed to simulate. Many topics in science, mathematics, and the social sciences do have areas where reasonable models do exist. Examples include ecological situations such as discussed above, chemical experiments where the reactions are too dangerous to do in class, physics simulations that allow students to change the laws of gravity within the computer, mathematics graphing packages to produce curves from various equations, and historical simulations such as "Oregon Trail" that simulate pioneers crossing the western United States.

It is important to note, however, that the amount of time and money needed to develop sinulations and games is very high. As a consequence, not as many good simulations and educational games are available as are drill and practice or tutorial programs. Furthermore, students must be cautioned that they are working with a model, not the real situation, and that not all the effects that can occur in the real world situation can be accounted for in the simulation. Finally, it cannot be stressed too strongly that simulations should not replace real life experiences, but should enchance and augment these experiences. The best way to learn a subject such as ecology is to involve students in the actual experiences of ecology, but multiple ecological experiences may not be available to all students in school. Ecological simulations can therefore be used to give students the opportunity to experience situations that they would not be able to otherwise experience.

Writing and Problem Solving Tools

The computer as a tool for teaching writing and problem solving is rapidly becoming more widespread. Word processing software now available has made it much easier for school systems to implement this use of computers. For example, the Bank Street Writer software (12) retails for less than $75.00 and provides a well designed word processing package for school age students. The package comes with a tutorial diskette that leads the student through the potentials of the program. Figure 7.8 shows a screen display from the tutorial diskette explaining how to Correct ~~insert~~ a word in~~to~~ text that has already been written.

Figure 7.8 About Here

Word processors such as Bank Street Writer do not provide as many functions as word processing systems for office use, but they allow students to create textual materials after a short training time. Proponents of the use of word processors indicate that students are able to make corrections, insertions and deletions of their textual material with much less effort than with conventional methods. Other benefits rslut from allowing students to build compound sentences and/or paragraphs by using teacher provided "kernel" sentences. Such uses allow students to produce and experiment with longer and more complex writings.

Other software packages that are not normally thought of as "educational" may also be utilized in problem solving situations. Spreadsheet packages, such as <u>VisiCalc</u>, can be used in mathematics classes as an aid to solving word problems. Spreadsheets allow formulas or numbers to be placed into the individual cells of the spreadsheet. Figure 7.9 shows a Visicalc screen that was set up to work on a problem of deciding how much carpet to buy for a number of rooms in a building. By changing the value of cell E2 ~~C3~~, where the cost per square yard is stored, the program changes the values of other cells based upon that cost. Figure 7.9 shows the new values. Note that only one single change was made to the sheet, but it resulted in new values being calculated and displayed in many cells.

Students can see the results of calculations without doing the arithmetic many times. However, they still must understand the equation that was used to set up the cells. The spreadsheet is actually being utilized as a computer language that is programmed to preform calculations for the student.

Figure 7.9 About Here

In addition to spreadsheets, programming languages -- especially BASIC -- are often utilized in mathematics classrooms as a tool to solve problems. For example, students may be given a programming problem such as "Write a program to calculate the area of a triangle based upon values that are inputted by the user of the program". The student must use knowledge of the computer language and of geometry to correctly solve the problem. Figure 7.10 shows a simple program that could be written for the IBM Personal Computer in BASIC to solve the problem. Line 80 does the actual calculation of the area based upon the values of the Base and Height provided by the user of the program. A key point is that the computer does not, on its own, solve the problem. The student programs the computer and thus must know the mathematical concepts and equations needed for the problem.

Figure 7.10 About Here

Questions to Consider

Why would this utilization, i.e. "Learning With Computers" be useful to students and teachers?

The reasons for use of this method fall into two major areas. First, as do most good teachers, we believe, that students need opportunities to learn by a variety of methods. Much research on what actually happens in classrooms seems to show a great deal of time being spent in "teacher talk", where the teacher is almost the sole source of information. The use of simulations, games, and tool software often changes the role of the teacher from "knowledge source" to facilitator and question provider. In addition, using these types of programs changes the pace of the class, allowing students to work in different instructional environments. Students appear to enjoy variety just as much as anyone does.

The second reason for utilizing "Learning With Computers", centers on the type of skills involved. As we have mentioned, it appears that students are doing types of tasks much different than in "Learning From the Computer". Instead of practicing previously learned skills, as in "learning from", students are involved in decision making processes while interacting with simulations and games, or utilizing tools with software such as word processors and spreadsheets. These type of skills are needed in real world situations and must be learned and practiced in schools to effectively be used after schooling has been completed.

When would this utilization best be considered?

There appear to be many places that the software types of this role would be useful. Most simulations and games will probably going to be used as extensions of topics or alternative methods of instruction in most content areas. For example, a simulation of a westward journey across America by pioneers offers an alternative way to study this topic as compared to traditional textbook and teacher lecture approaches. Some computer games might, however, be used to replace practice sessions on arithmetic skills, replacing worksheets or drill and practice computer software. While there is not enough good software in this area it does cover a wide range of grade levels. The content area chpaters found later in this book provide recommendations on specific software and on organizations that may be useful in providing software for schools.

What kinds of skills does the utiliztion provide?

The "learning with computers" utilization has the potential to develop a wide range of skills. Some games may only sharpen low level skills such as arithmetic and spelling. Other games and simulations require higher level evaluation, synthesis, and analysis skills. The tools are more directed towards providing students with opportunities to use information in different ways and to use information in different ways and to confront more possibilities than without the computer.

IV. LEARNING ABOUT THINKING WITH COMPUTERS

Learning about thinking with computers is probably the most unconventional, but at the same time perhaps the most powerful of the various uses presented in this volume. In this role, the computer is used to help students develop new patterns of thinking that may assist them in many different learning situations.

The main proponent and artful presenter of this role of the computer is Dr. Seymour Papert of the Massachusetts Institute of Technology's LOGO Laboratory. Papert is a self-confessed educational radical when it comes to his ideas of how computers and children should interact, and his underlying philosophy is based upon the Swiss psychologist Jean Piaget.

Papert, while greatly influenced by Piaget, extends and modifies some of Piaget's ideas in his own model of the interaction of computers and children. In its simplest form, Papert's model expresses the idea that the computer can be used to "concretize" many of the formal learning situations presented to students in classroom environments. He notes in his book Mindstorms (13), that our culture is very rich in some types of learning situations such as the concept of "a pair" (e.g., a pair of socks or a pair of shoes) but very poor in situations that require or develop structured thinking. Structured thinking involves the process of dividing a large problem into smaller parts or components that can be easily handled on an individual basis and then combined to give a solution to the original larger problem.

One way Papert sees to facilitatea the development of structured thinking is through the interaction of the student with the computer by means of a computer language that is itself structured. Such a computer language should allow the writer of a program to solve a relatively complicated problem by breaking the problem down into smaller and simpler components. That is, by constructing "procedures" or sub-programs to produce a total program that solves the problem. Since BASIC was not developed to have this kind of structure, Papert and others have criticized it as an inappropriate first language for someone learning about computer programming, especially children. Papert and colleagues propose the LOGO language as an alternative.

As we saw in chapters 5 and 6, LOGO and Basic have many differences in the way commands are presented and in the total structure of programs prepared under both languages. Nevertheless, the question of one language being "better" than another, especially when it comes to the area of developing problem solving skills, is one that is not easily answered. The next chapter will deal with area of problem solving in more detail. Thus, much of our discussion in this area will be deferred until then. For present purposes, it is sufficient to note that programming computers seems to place different demands upon students' thinking than do many current school situations. The student is not asked to present the teacher with information in a form similar to the teacher's presentation, but to create new programs based upon rules of the computer language. These creative skills are not ones that are often used by students, and we as educators may need a new set of instructional methods to "teach" them properly. Whether these skills of "rule use" will transfer back to standard curricular areas is an area of current research. At present it is not known whether the answer to this question will be positive or negative.

Question to Consider

Why would this utilization be useful to students and teachers?

If Papert and his supporters are correct this utilization of the computer could open new possibilities for students. Exploration of student generated computer environments, sometimes called "microworlds", could be a powerful force for student learning. It is too early to say if real changes in student thinking will take place. However, one thing appears to be true: Students are highly motivated to work with LOGO and LOGO type situations when they are implemented in schools. This higher motivation is in itself a payoff for the consideration of these type of activities. Continued research may give us the answers to our questions regarding thinking skills and transfer to other tasks.

When would this utilization best be considered?

LOGO environments are often talked of in terms of no boundaries. Students from kindergarten to high school have successfully worked with this language. It would require significant teacher training to prepare teachers to work with students in a manner consistent with the philosophy of the program. It is not an "open the package and turn on the machine" type of environment, although the students usually rapidly learn to use the language.

What kinds of skills does the utilization provide?

As has been mentioned, the hoped-for-benefits of these environments involve changes in the way students think about problems, changes that can be transfered to other situations. At a general level, computer programming involves a situation where the student manipulates the computer rather than one in which the computer manipulates the student.

V. MANAGING LEARNING WITH COMPUTERS

In contrast with the other uses of the computer in education that have been discussed, this use of the computer is indirectly related to student learning. That is, in this role the computer is used to help the teacher and/or administrator with the paperwork and organizational tasks that often take time away from actual teacher-student interaction.

There are several ways in which an individual classroom teacher can make use of the computer in this role. First, the computer can aid in keeping track of grades and in giving desired information about these grades such as averages and totals. For teachers with large numbers of students this can yield a large savings in time. There are several avenues by which teachers can obtain programs that would do this type of task for them. There are commercially available programs for this type of application but teachers may find that locally produced programs are more useful for their particular situation. In fact, some schools use students in computer classes to write simple programs for this use. Not only does the school get a customized program, but the students can work on a "real world" example for a project.

The second major use of this application is in using the computer to generate test questions. These can be for the class in general or to construct special remedial questions for individual students based on their needs. This does involve a significant amount of time for the teacher or assistant in inputing the test questions for the first time. After that, however, the teacher can easily pick out certain items or groups of items to use for tests. In addition, if the items have been coded in terms of objectives, teachers can pick out items for particular objectives. If multiple items have been entered for each objective teachers can randomly choose items from those available. In this manner, students that have missed an exam or need to repeat a quiz will have a unique test.

Third, the computer can be used to keep track of attendance or other student records, such as academic progress or achievement test scores. This is probably most useful at the school level. Administrators can compile teacher reports, using the computer, to produce school and district information about student attendance, achievement levels, and special needs.

A fourth area is that of communication between computers in a school or communications with machines at other locations away from a school. As was described in Chapter 3, there are methods by which computers can communicate with one another. Within a school, the computers in a computer lab might be joined in a network where the individual computers can communicate with each other and with a primary machine that might have more storage space or a printer attached. Such networking offers the advantage of sharing expensive components, such as hard disks and printers with many machines. Instructionally, the teacher will also have the ability to send a program to all of the computers in the lab instead of loading it one at a time into each machine.

Being able to communicate with other computers away from the school setting also offers some advantages. Some software companies are now making their products available through large computers which can send the program to a microcomputer over a telephone line. The microcomputer can then save these programs on a diskette for future use. Teachers and administrators can also use such communications capability to interact with other professionals who are using technology in school settings. Some of the commercial information services such as "The Source" and "Compuserve" allow users to leave messages or communicate over a open "bulletin board" that allows for interaction among users. While some costs are involved (commercial services charge for the time the user is connected to the host computer) it does offer an interesting way to obtain useful information. As infomation and data bases become more easily used, and more readily available, students may also begin to use them as a resource similar to reference books. The system could be queried about a certain topic, such as "Volcanoes", and the user could obtain specific information about topics such as number of active volcanoes, major eruptions, and perhaps information about how volcanoes are produced.

Finally, the computer can be used by school administrators for many of their administrative duties. Class scheduling, budgets, availability and qualifications of substitute teachers, supply inventories, and similar tasks are examples. In some schools the same computers have been used both for direct learning situations and for these types of administrative tasks. Careful planning is needed, however, to allocate the computer resources to best benefit the students. Chapter 16 of this text will discuss these administrative uses of computers in more detail.

REFERENCES

1. Taylor, Robert (Ed.) The Computer in the School: Tutor, Tool, Tutee. New York: Teacher's College Press, 1980.

2. Luehrman, A. "Microcomputers and Children." A Paper presented at The Wingspread Conference on Microcomputer Education, Feb. 1982.

3. Luehrman, A. and Peckham, H. Computer Literacy: A Hands-on Approach. New York: McGraw-Hill, 1982.

4. Speed Drill. St.Paul, MN: Minnesota Educational Computer Consortium, 1982.

5. Brown, J. and Burton, R. "Diagnostic Models for Procedural Bugs in Basic Mathematical Skills". Cognitive Science, Vol. 2, pp. 155-192, 1978.

6. Suppes, P. "Observations About the Application of Artificial Intelligence Research to Education" in Walker,D. and Hess, R. Instructional Software: Principles and Perspectives for Design and Use. Belmont, CA.: Wadsworth, Inc., 1984.

7. Malone,T. "Towards a Theory of Intrinsically Motivating Instruction", Cognitive Science, Vol. 4, pp. 333-369, 1981.

8. Bork, A. Learning With Computers. Bedford, MA.: Digital Press, Inc., 1981.

9. Plant Growth. Staten Island, NY: Classroom Consortia Media, Inc., 1984.

10. Levin, W. "Interactive Video: The State-of-the-Art Teaching Machine". The

Computing Teacher, Vol. 11, No. 2, pp 11-17, 1983.

11. *Odell Lake*. St. Paul, Mn.: Minnesota Educational Computer Consortium, 1982.

12. *Bank St. Writer School Edition*. New York: Scholastic, Inc., 1983.

SUGGESTED READINGS

Bork, A. "Computers and Learning: Don't Teach BASIC". *Educational Technology*, 22(4), pp. 33-34, 1982.

Corburn, P. et. al. *Practical Guide to Computers in Education*. Reading, MA: Addison Wesley, 1982.

Harper, D. and Stewart, J. (Eds.) *Run: Computer Education*. Monterey, CA: Brooks/Cole, 1983.

Hunter, B. *My Students Use Computers*. Reston, VA: Reston, 1984.

Luehrmann, A. "Don't Feel Bad About Teaching BASIC". *Electronic Learning*, Sept., 1982, pp. 23-24.

QUESTIONS FOR DISCUSSION

1. Place yourself in the role of a Junior High School Science Teacher who has been asked to develop some ideas on how computers might be used in the teacher's science class. List utilizations that would be most appropriate for the science teacher to consider. Defend your choices.

2. Many schools have only a very limited number of computers per school (one or two). Which of the methods suggested in this chapter would be most appropriate and practical for such a setting?

3. Rank the various methods of utilization in this chapter as to the amount of teacher training needed to implement.